(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
18 September 2003 (18.09.2003)

PCT

(10) **International Publication Number**
WO 03/077423 A2

(51) **International Patent Classification**[7]: **H03M**

(21) **International Application Number:** PCT/US03/06774

(22) **International Filing Date:** 7 March 2003 (07.03.2003)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
60/362,721 8 March 2002 (08.03.2002) US

(71) **Applicant: QUELLAN, INC.** [US/US]; 250 14th Street, N.W., Atlanta, GA 30318 (US).

(72) **Inventors: HIETALA, Vincent Mark**; 6200 Eubank Blvd., N.E., #528, Albuquerque, NM 87111 (US). **KIM, Andrew Joo**; 3645 Peachtree Road, #312, Atlanta, GA 30319 (US).

(74) **Agent: WIGMORE, Steven P.**; King & Spalding LLP, 191 Peachtree Street, Atlanta, GA 30303-1763 (US).

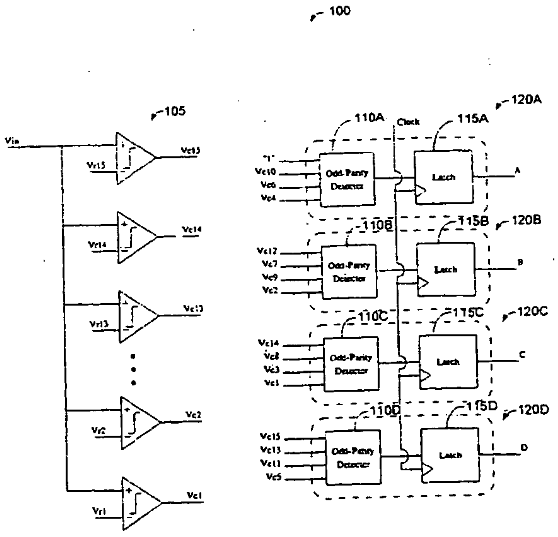
(81) **Designated States** *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Designated States** *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) **Title:** HIGH-SPEED ANALOG-TO-DIGITAL CONVERTER USING A UNIQUE GRAY CODE



(57) **Abstract:** A method for high speed communications uses an inventive Q-Gray code. The Q-Gray code simplifies the hardware needed to convert analog Q-Gray code signals to digital signals. An analog-to-digital converter can use a plurality of comparators for receiving the multilevel signal and a plurality of decoder blocks coupled to comparators for decoding the multilevel signal. Each decoder block can include an equal number of inputs. Specifically, each decoder block can also include a parity detector with an equal number of inputs. Each decoder block can also employ a bank of identical parity detectors relative to another decoder block. Each comparator of the analog to digital converter can have an individually or externally adjustable (or both) threshold level.

WO 03/077423 A2

HIGH-SPEED ANALOG-TO-DIGITAL CONVERTER USING A UNIQUE GRAY CODE

STATEMENT REGARDING RELATED APPLICATIONS

The present application claims priority under 35 U.S.C. § 119(e) to provisional patent application entitled, "HIGH-SPEED ANALOG-TO-DIGITAL CONVERTER USING A UNIQUE GRAY CODE," filed on March 8, 2002 and assigned U.S. Application Serial No. 60/362,721, the entire contents of which are hereby incorporated by reference.

FIELD OF THE INVENTION

The present invention is directed to analog-to-digital (ADC) converters and, more particularly, a very high-speed ADC is described for the level detection of, or equivalently, the data extraction from high-speed multilevel (ML) waveforms.

BACKGROUND OF THE INVENTION

Complex modulation schemes are desirable to obtain increased spectral efficiency and, therefore, increased data throughput for high-speed communications systems. The extraction of data from the multi-Giga symbol per second (Gsym/s) ML waveforms can be difficult, but is critical for effective implementation of a high-speed communications system.

Conventional flash ADC's are common in high-speed systems; however, these ADC's suffer three shortcomings. First, they are based on conventional Gray codes that result in asymmetric decoder circuitry in the sense that the circuitry for decoding the simplest bit channel can be significantly more complex than the circuitry for the most complex bit channel.

Conventional Gray codes are described in U.S. Patent 2,632,058, entitled "Pulse Code Communication", which issued to Gray on March 17, 1953. This reflected-binary code (now generally called a Gray code) described in the '058 patent provides for improved performance of pulse coded communications based on encoding data in time vs. voltage.

With conventional Gray code analog to digital decoders, buffers are usually used to delay the result from the simplest bit channel to match the output delay of the most complex channel. The overall result is an ADC that has high complexity which translates into high power consumption and lower achievable speed.

The second shortcoming of flash ADC's based on conventional Gray codes is that the error rate on each of the bit channels is skewed. This skew can reduce the performance of error correction mechanisms where it is assumed each bit channel has the same error probability.

The third shortcoming of conventional flash ADC's is that they have uniformly spaced decoding thresholds. In many communication contexts, the received signal is distorted by signal dependent noise and the use of uniformly spaced thresholds is suboptimal.

In view of the foregoing, there is a need in the art for an ADC where the thresholds are independently and externally adjustable to provide a means for maintaining optimal decoding in the presence of (possibly time-varying) signal dependent distortions. There is a further need in the art for an ADC that can support the adjustable quantization of ML signals within the receiver of a high-speed telecommunication system. Another need exists in the art for an evenly distributed bit error rate across each of the bit channels in a multilevel signal. And further, a need exists in the art for a simpler ADC design for high speed communications.

SUMMARY OF THE INVENTION

A design for an analog-to-digital converter that can decode a unique multilevel Gray code can be made more simple than conventional analog to digital converters. More specifically, because of certain properties of the inventive and exemplary Q-Gray codes of the present invention, the analog-to-digital converter can comprise a plurality of comparators for receiving the multilevel signal and a plurality of decoder blocks coupled to comparators for decoding the multilevel signal. Each decoder block can comprise an equal number of inputs. Specifically, each decoder block can also comprise a parity detector with an equal number of inputs. Each decoder block can also comprise a bank of identical parity detectors relative to another decoder block.

According to another exemplary aspect of the present invention, in one exemplary embodiment, each comparator of the analog to digital converter can have an individually adjustable threshold level. Further, each comparator can also have an externally controllable threshold level. With such individually adjustable thresholds, the comparators can process a multilevel signal comprising non-uniformly spaced decoding thresholds.

With this simple design, the analog to digital converter can reduce dissipated power and can increase achievable operational speeds for communications. Some unique properties of the Q-Gray coded multilevel signal that dictate the design of the aforementioned analog to digital converter are the following: The maximum number of bit-toggles incurred in a bit channel while sequentially traversing the code is minimized. An example of a situation in which the above minimum-maximum criterion is satisfied is when the Q-Gray code has a transition density that is maximally evenly distributed among bits in the binary representation of the multilevel signal. This maximally even transition distribution results in the property that the bit error rate is substantially evenly distributed across each bit channel.

The present invention provides for a method of high speed communications using an inventive Q-Gray code. In turn, the Q-Gray code simplifies the hardware needed to convert multilevel Q-Gray coded signals to binary signals.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A is a block diagram of an exemplary 4-bit analog-to-digital converter (ADC) according to one exemplary embodiment of the present invention.

Figure 1B is a block diagram of an exemplary 3-bit ADC according to one exemplary embodiment of the present invention.

Figure 1C is a block diagram of an exemplary 5-bit ADC according to one exemplary embodiment of the present invention.

Figure 1D is a block diagram of an exemplary 4-bit ADC according to one exemplary embodiment of the present invention.

Figure 2 is a top-level schematic circuit for the 4-bit ADC shown in Figure 1D according to one exemplary embodiment of the present invention.

Figure 3 is a schematic circuit diagram of the VC4b block of Figure 2 according to one exemplary embodiment of the present invention.

Figure 4 is a schematic circuit diagram of the VC3b block of Figure 2 according to one exemplary embodiment of the present invention.

Figure 5 is a schematic circuit diagram of the input buffer, REFBUF, of Figure 2 according to one exemplary embodiment of the present invention.

Figure 6 is a schematic circuit diagram of the IREF circuit block of Figure 2 according to one exemplary embodiment of the present invention.

Figure 7 is a schematic circuit diagram of the comparator block, COMP, of Figure 3 and Figure 4 according to one exemplary embodiment of the present invention.

Figure 8 is a graphical depiction of the simulation results for the exemplary 4-bit ADC shown in Figure 2 according to one exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

The use of an innovative "Q-Gray code" described herein allows for simplification of the decoder of flash-based ADC's and in a faster ADC design. As an additional benefit, the Q-Gray code is also an optimal code for ML communication systems. The general class of Q-Gray codes has the unique property of allowing for a corresponding flash ADC to have minimal decoder complexity and therefore maximum conversion speed with minimal power consumption.

The Q-Gray codes of the present invention are a subset of the large class of Gray codes. Specifically, the Q-Gray codes of the present invention are Gray codes with the following property: letting M_b denotes the maximum number of bit toggles incurred in bit position b as the code is incremented from 0 to 2^N-1 and followed by 0 (where N is the number of bits comprising the code), a Q-Gray code achieves the minimum possible value of $\max_b\{M_b\}$, (i.e. the largest number of toggles incurred over any bit position is minimized). Thus, a Q-Gray code distributes the number bit toggles (from adjacent level transitions) as evenly as possible over all the bit positions.

This defining property of Q-Gray codes minimizes the maximum number of aggregate number of bit toggles (on adjacent level transitions) on any bit position. In other words, for the inventive Q-Gray codes, the maximum number of bit-toggles incurred in a bit channel while sequentially traversing the code is minimized. This not only allows for simple bit decoders within the ADC, but also offers the total bit error rate to be approximately 1/$\log_2$N of the symbol error rate (as with any Gray code) <u>and</u> an approximately equal error rate for each bit of the code words (which does not hold for most Gray codes).

In order to understand the impact of coding upon an ADC design, Table 1 lists several possible coding candidates, one of which is according to an exemplary embodiment of the present invention. As in Table 1, much of this document will use N=4-bit coding (i.e. 16-level modulation) as an example; however, the extension to other numbers of bits will be clear to those skilled in the art and are included as part of the present invention. Table 1 presents a conventional binary code, a conventional Gray code, and one exemplary Gray code of the present invention that has been labeled as a "Q-Gray" code.

Table 1. Comparison of several possible 4-bit ML codes.

Level	Binary	Gray Code	Q-Gray Code
15	1111	1000	0000
14	1110	1001	0001
13	1101	1011	0011
12	1100	1010	0010
11	1011	1110	0110
10	1010	1111	0111
9	1001	1101	1111
8	1000	1100	1011
7	0111	0100	1001
6	0110	0101	1101
5	0101	0111	0101
4	0100	0110	0100
3	0011	0010	1100
2	0010	0011	1110
1	0001	0001	1010
0	0000	0000	1000

In order to appreciate the features of the Q-Gray code, it will be useful to first review some background material. First, a symbol for a simple ML communication system is simply an amplitude level. A symbol error occurs when a received symbol is different than the symbol that was transmitted. A bit error occurs when a received data bit is different than the transmitted data bit. Note that in general, a data error(s) will occur when a symbol error occurs, but the number of data errors which occur depends on the nature of the error as well as the way that the data bits represent the symbol (i.e. the coding used).

For a ML system, almost all errors result from the case where a received symbol is wrongly detected as an adjacent symbol level. Therefore, the coding used should minimize the number of bits that will be in error if a symbol is in error by one level. The traditional Gray-code does exactly this, by having the code words for any pair of adjacent levels differ in only one bit position.

In order to further explain this, define the following:

$P(err_s)$ is the symbol error probability (or rate),

$P(err_b)$ is the total bit error probability (or rate),

and

$P(err_{b=A})$, $P(err_{b=B})$, $P(err_{b=C})$, and $P(err_{b=D})$ are the bit error probabilities (or rates) of bit *A, B, C,* and *D,* respectively.

The following discussion will also use the standard notation for conditional probabilities where *P(X|Y)* means "the probability of" *X* "given" *Y*. Multiple conditions will be separated by ","'s as in *P(X | Y,Z)* means the probability of *X* given *Y and Z*.

For the following discussions, in regard to any given code, the bits will be termed A, B, C, and D, in sequence from left to right with A being the left-most bit and D being the right-most bit. The transmitted level will be referred to as *L* and the received level as $\hat{L}$. We will assume that each symbol is equally likely to be transmitted, *i.e.*, *P*(*L*)=1/16 where *L*=0,..,15 is the symbol transmitted. The vast majority of communications channels have noise characteristics in which smaller noise perturbations are more likely than larger perturbations, *e.g.*, Gaussian noise. In such situations, errors in detecting the symbol are almost always due to declaring a level adjacent to the one transmitted, i.e. declaring $\hat{L} = L - 1$ or $\hat{L} = L + 1$. If optimal detection is used, then the probability of these two types of errors is the same, *i.e.*, $P(\hat{L} = L - 1|L) = P(\hat{L} = L + 1|L)$.

If we additionally assume that optimal transmit level spacing is used, then the above error probabilities are independent of the transmit level, as shown in Equation (1):

$$\frac{1}{2}P(err_s) = P\left(\hat{L} = L-1 \middle| L\right) = P\left(\hat{L} = L+1 \middle| L\right) \tag{1}$$

for all L except in the degenerate case where $\hat{L} = -1$ and $\hat{L} = 16$ for which the probability of these types of errors is zero since the receiver only makes detections corresponding to valid symbols.

The bit error probability can now be shown to approximately be the symbol error probability multiplied by a scalar that is a characteristic of the particular binary code used to represent the symbols. This scalar multiplier corresponds to the average number of bit toggles incurred as the binary representation is sequentially traversed. To be more specific, define the bit toggle counting function $\tau(L,L+1)$ to be the total number of bits toggled in transitioning from L to $L+1$ (or vice-verse). For example, in the traditional binary representation, $\tau(3,4)=4$ since $3=0111_b$ and $4=1000_b$ differ in all four bit positions. It will also be useful to define the function $\varphi(L,L+1)=\tau(L,L+1)/N$ which is the fraction of bits toggled in transitioning from level L to $L+1$. The bit error probability can be determined by using these functions.

By the definition of conditional probabilities and the independent error rates of each level, the total bit error probability can be expressed in terms of levels, L, as shown in Equation (2):

$$P\left(err_b\right) = \sum_{L=0}^{15} P(err_b \mid L)P(L) \tag{2}$$

Again applying the definition of conditional probabilities, Equation (2) can be further expanded in terms of errors occurring to the level above or below, as shown in Equation (3):

$$\begin{aligned} P\left(err_b\right) = P(L)\sum_{L=0}^{15}\Big[&P(err_b \mid L, \hat{L} = L-1)P(\hat{L} = L-1 \mid L) + \\ &P(err_b \mid L, \hat{L} = L+1)P(\hat{L} = L+1 \mid L)\Big] \end{aligned} \tag{3}$$

The total bit error conditional probabilities in Equation (3) are in fact equal to the previously defined average bit toggle function, φ. This can be understood by considering, for example, the first term $P(err_b \mid L, \hat{L} = L-1)$. This expression means, "the probability of a bit error given the transmitter level is L and the received level,

$\hat{L}$, is L-1". With the givens, a symbol error has occurred and the resulting probability will be 1/4, 2/4 or 3/4 depending on how many bits toggled erroneously (1, 2, or 3; respectively) due to detecting the level as L-1 vs. L. This is precisely the definition of φ. Therefore,

$$P(err_b \mid L, \hat{L} = L-1) = \varphi(L-1, L) \qquad (4)$$

and

$$P(err_b \mid L, \hat{L} = L+1) = \varphi(L, L+1) \qquad . \qquad (5)$$

Equations (4) and (5) can now be substituted into Equation (3), yielding Equation (6), as shown below:

$$P(err_b) = P(L)\sum_{L=0}^{15}\left[\varphi(L-1,L)\,P(\hat{L} = L-1 \mid L) + \varphi(L,L+1)\,P(\hat{L} = L+1 \mid L)\right] \qquad (6)$$

The bit error rate can now be expressed in terms of the symbol error rate by applying Equation (1), as shown below, to generate Equation (7):

$$P(err_b) = \frac{1}{2}P(err_S)P(L)\sum_{L=0}^{15}\left[\varphi(L-1,L) + \varphi(L,L+1)\right] \qquad (7)$$

Since $\varphi(-1,0)$ and $\varphi(15,16)$ are zero, we can rewrite Equation (4) to provide the new Equation (8):

$$P(err_b) = \frac{1}{2}P(err_S)P(L)\sum_{L=0}^{14}2\,\varphi(L,L+1) \qquad . \qquad (8)$$

A bit of trivial simplification gives the desired result in Equation (9):

$$P(err_b) = P(err_S)\frac{1}{16}\sum_{L=0}^{14}\varphi(L,L+1) = P(err_S)\frac{1}{4*16}\sum_{L=0}^{14}\tau(L,L+1) \qquad . \qquad (9)$$

Thus, as previously mentioned, the bit error probability is approximately the symbol error probability scaled by the average number of fractional bit toggles as the binary representation is sequentially traversed. It is not exactly the average due to the normalization factor of 16 (instead of 15) where the inconsistency arises due to boundary effects at the ends of the sequence. The scale factor (*i.e.*, the summation in Equation (9)) is a property of the code used to represent the symbols, and is minimized by Gray-codes where each summand is the minimum value of 1 (only 1 bit toggles per level transition).

The definitions of $\tau(L,L+1)$ and $\varphi(L,L+1)$ may be adapted to also provide the bit error probability for a given bit $X=A, B, C,$ or D. To do this, we simply define

$$\tau_X(L,L+1)=\begin{cases}0:\text{if bit X of L and L + 1 are the same}\\1:\text{if bit X of L and L + 1 are different}\end{cases}$$

(10)

Then, using τ_X in place of τ in Equation (9) gives the bit error probability for bit X, *i.e.*, $P(err_{b=X})$. These values are tabulated in Table 2 for the conventional binary, conventional Gray, and Q-Gray codes.

Table 2. Normalized bit error rates for several exemplary coding examples.

Code	Bit Error Rate (normalized to the symbol error rate)				Total Bit Error Rate
	A	B	C	D	
Binary	1/64	3/64	7/64	15/64	26/64
Gray Code	1/64	2/64	4/64	8/64	15/64
Q-Gray Code	3/64	4/64	4/64	4/64	15/64

From Table 2, it is seen that the Gray codes give a total bit error rate of somewhat better than 57% of a straight binary code. While the error rate analysis is present for N=4 bits, those skilled in the art will recognize that everything carries over to arbitrary values of N. In particular, Eq. (9) can be generalized to be

$$P(err_b)=P(err_S)\frac{1}{M}\sum_{L=0}^{M-2}\varphi(L,L+1)=P(err_S)\frac{1}{N*M}\sum_{L=0}^{M-2}\tau(L,L+1)$$

(11)

where $M=2^N$ is the number of levels.

The distribution of bit errors within the individual bits (labeled A, B, C, and D) is almost ideally distributed by the use of the Q-Gray Code, thereby demonstrating a significant advantage over the conventional Gray Code. While this uniform distribution of the error over the bits is advantageous in that it eliminates the need for customized processing to account for skewed error rates, the largest benefit of this uniform distribution is that it allows for an improved realization of a flash ADC. In order to understand this benefit, the basic operation of each necessary bit decoder needs be explored.

The ADC is assumed a flash type converter with M-1=15 comparators (in the case of 16-level or 4-bit converters). These 15 comparators are referenced with 15 reference voltages that set the decision point between the expected symbol levels. Therefore, in general, they will be positioned mid-point (perhaps scaled according with the noise variances) between the ideal level voltages. Note that implementation of any of the exemplary codes will take M-1 comparators – this is a fundamental property of a flash converter.

A given bit decoder can be viewed as asserting its output based on the relationship of the input voltage as compared to the bit toggle thresholds required for a given bit. That is, if a given bit has say, for example four bit toggles, over the symbol input range, a four-input conventional odd-parity function gate connected to the appropriate four comparators is all of the hardware that's required to decode the bit.

Referring back to Table 1, we note that the numerator of each cell in the middle four columns is the summation shown below in Equation (12):

$$\sum_{L=0}^{M-2} \tau_X(L, L+1) ,$$

(12)

which counts the total number of times bit X is toggled as symbols are sequentially traversed. For example, bit A of traditional binary coding will only require a one-input odd-function gate (i.e. a buffer or no logic) to be the implemented. On the other hand, bit D of a traditional binary coding, will required a 15-input odd-function gate to bit decode – a large level of complexity that will undoubtedly be difficult to build at high speed. The traditional Gray code represents a significantly improved situation, but for example, bit D still requires an 8-input odd-function logic gate.

In contrast, the Q-Gray code has a minimum-maximum number of bit toggles per bit with 3 bits having 4 toggles and 1 bit having 3 toggles. Therefore, each bit requires at most a 4-input odd-parity logic gate for bit decoding. As will be seen below, a 4-input odd-parity gate can be implemented with 3 standard exclusive OR (XOR) gates.

Since the number of inputs to an odd-function gate will directly impact the gate's speed, the Q-Gray code represents a significant increase in ADC speed.

It is believed that the exemplary Q-Gray codes of the present invention are unique and that they exist for at least N =2, 3, 4, and 5 with larger searches being restricted by computer resources. In order to explore this further, the case of N=4 is considered in more detail, as shown in Table 3.

Table 3. Exemplary 4-bit Q-Gray Code.

	Bit			
Level	A	B	C	D
0	1	0	0	0
1	1	0	1	0
2	1	1	1	0
3	1	1	0	0
4	0	1	0	0
5	0	1	0	1
6	1	1	0	1
7	1	0	0	1
8	1	0	1	1
9	1	1	1	1
10	0	1	1	1
11	0	1	1	0
12	0	0	1	0
13	0	0	1	1
14	0	0	0	1
15	0	0	0	0

The search for a Q-Gray code can be a very challenging endeavor if done by brute force. The 16 values of an N=4 (4 bit word) can be selected in 16 factorial ways (20,922,789,888,000 possibilities), which is a somewhat restrictively large number for a computer search unless care is taken with the search algorithm used.

For example, if 10 million candidates are tested per second (optimistic workstation search speed), it would still take 24 days for even this simple N=4 search to complete. Fortunately, if the problem is cast into the underlying *bit transition operator lists* (BTOL's), there are only 4^{15} possible trial lists (1,073,741,824 possibilities).

A BTOL is a list of sequential bit transition operations required to produce a valid Gray code. A BTOL inverts the indicated bit of the binary word. In particular, if the Kth element of the list is bit A, then that means the binary representation of the decimal number K is obtained by taking the binary representation of K-1 and inverting the "A" bit (and similarly for the other bits). If a list results in a non-repeating sequence of binary words, it represents a valid Gray code.

Furthermore, the resulting sequence of binary words represents a Gray code regardless of the seed word. Because Gray codes are produced by BTOL's, one can search among valid BTOL's (instead of the much larger set of sequences of binary words) for a Q-Gray code as previously stated. This search could be significantly further reduced by symmetry arguments and other simple rules, but this is a manageable search space for low-end computers and was selected as the starting point.

A computer search resulted in 5712 BTOL's that produce Gray codes. These operator lists can be "seeded" with 16 possible starting levels (represents all possible bit inversions) plus 4! ways to arrange the bit columns, resulting in a total number of "Gray" codes of 2,193,408.

This list of 5712 possibilities was further filtered to find desired "optimal" Q-Gray codes which minimize the maximum number of transitions per bit. There must be 15 transitions total that are divided between the 4 bits. This results in the optimal number of transitions per bit being 4 with one bit having 3. This search resulted in 384 operator sequences.

These 384 sequences were all observed to be based on a single basic bit-transition operator sequence:

CBCADABCBADBDCD

(13)

and with operator A returning the original "seed" from the last element in the resulting code. The sequence in Eq. 13 produces the Q-Gray code listed in Table 3 when a seed of "1000" is used.

Since any column operator can be interchanged with another, and the sequence can be started at any position, the total number of sequences is 4! * 16 = 384, which is correctly identical to the search result. Note that the sequence can be reversed, but in fact, reversal of the sequence only makes the sequence with a bit interchanged (therefore accounted for in the 4! term). Finally, since again, there are 16 possible seeds for the operator lists, there are a total of 384*16 = 6,144 rather good Q-Gray code sequences. It appears that all 6,144 of these possible codes are unique, but are really just forms of one another.

The N=2 case is trivial as the only Gray Code is a Q-Gray code. Additional computer searches found Q-Gray codes for the 3 and 5 bit (8 and 32 level) resolutions. The exemplary Q-Gray code for the N=3 bit case is given by the BTOL:

ABCBABC

(14)

with operator B returning the original seed from the last word in the code. Table 4 gives the Q-Gray code resulting from this BTOL when a binary seed value of "000" is used.

The exemplary Q-Gray code for the N=5 bit case is given by the BTOL:

ABACABADABEBDCDEBEDCACDECEBDCDE

(15)

with operator D returning the original seed from the last word in the code. Table 5 gives the Q-Gray code resulting from this BTOL when a binary seed value of "00000" is used.

It is believed that a Q-Gray code exists for higher resolutions (N>5) but are limited in search due to finite computer resources (the brute force search scales by (2^N)!, which quickly explodes and other methods also scale poorly). Further, additional unique Q-Gray codes for each of the three bit, four bit, and five bit examples discussed above can be derived by cyclically rotating the bit translation operating list to yield alternate and inventive Q-Gray codes. Similarly, additional unique Q-Gray codes can be derived from the three, four, and five bit examples mentioned above by interchanging bit channel assignments.

Table 4. Exemplary 3-bit Q-Gray Code

	Bit		
Level	A	B	C
0	0	0	0
1	1	0	0
2	1	1	0
3	1	1	1
4	1	0	1
5	0	0	1
6	0	1	1
7	0	1	0

Table 5. Exemplary 5-bit Q-Gray Code.

	Bit				
Level	A	B	C	D	E
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	0	0	0
3	0	1	0	0	0
4	0	1	1	0	0
5	1	1	1	0	0
6	1	0	1	0	0
7	0	0	1	0	0
8	0	0	1	1	0
9	1	0	1	1	0
10	1	1	1	1	0
11	1	1	1	1	1
12	1	0	1	1	1
13	1	0	1	0	1
14	1	0	0	0	1
15	1	0	0	1	1
16	1	0	0	1	0
17	1	1	0	1	0
18	1	1	0	1	1
19	1	1	0	0	1
20	1	1	1	0	1
21	0	1	1	0	1
22	0	1	0	0	1
23	0	1	0	1	1
24	0	1	0	1	0
25	0	1	1	1	0
26	0	1	1	1	1
27	0	0	1	1	1
28	0	0	1	0	1
29	0	0	0	0	1
30	0	0	0	1	1
31	0	0	0	1	0

Exemplary embodiments of the present invention will hereinafter be described with reference to the drawings, in which like numerals represent like elements throughout the several figures.

An exemplary embodiment of a 4-bit ADC 100 based on the Q-Gray code in Table 3 is illustrated in Figure 1A. Figure 1A is a block diagram of a flash converter 100 with the normal 2^N-1 (=15) input comparators 105 with threshold voltages (labeled Vr_i) set between the expected input symbol levels. It is noted that conventional flash ADC's use thresholds that are uniformly spaced in a set decoding range.

The present invention, in contrast, allows fully independent and externally adjustable control of the threshold voltages Vr_i. This flexibility provides more freedom in the signal decoding. Such flexibility is needed in communication settings where one has signal dependent noise or distortions and the optimal decoding thresholds are not uniformly spaced.

The comparator outputs are appropriately routed to the necessary bit decoders 110 of decoder blocks 120 to realize the desired coding. The decoder blocks 120 further comprise latches 115. Because the Q-Gray code of this exemplary embodiment is a Gray code, each comparator only drives one decoder input. This one-to-one assignment results in low gate loading and hence high speed as with all Gray-coded converters.

The one-to-one assignment between comparator outputs and decoder inputs is directly obtained from the BTOL and seed word for the Gray code. In particular, the inputs to a parity-detection decoder block 110 are (i) the comparator outputs corresponding to the list locations of the bit in the BTOL and (ii) the seed value for that bit.

Since the exemplary ADC 100 of Figure 1A uses the Q-Gray code, the 4 necessary bit decoders 110 are simple 4-input odd-parity functions as discussed above. And as previously stated, the inputs to the decoders 110 are taken from the BTOL and seed value.

For example, the A-bit decoder 110A takes as inputs logical 1 (from the seed value) and comparator outputs Vc4, Vc6, and Vc10 since A occurs in the BTOL sequence in Eq. 13 at the fourth, sixth, and tenth positions. Similarly, the B-bit decoder 110B takes Vc2, Vc7, Vc9, and Vc12 as inputs because the B operator occurs in the second, seventh, ninth, and twelfth positions in the BTOL in Eq. 13.

Furthermore, because the seed value is "0" for the B-bit in Table 4, it is unnecessary to provide this input to the odd-parity decoder block. Input assignments for the C and D-bit decoders 110C, 110D are similarly obtained from the BTOL. The 4 outputs of the bit decoders 110 are finally latched by the clock signal at a desired sample time.

The distribution of toggles as the code is traversed (i.e. the number of occurrences of a bit operator in the BTOL) directly impacts the structure of any Gray code decoder. The advantage of the even bit error distribution provided by the Q-gray code of the present invention can now be explained.

The complexity of a decoder circuit is determined by the number of gates in the circuit which increases linearly with respect to the number of inputs to the parity detection circuit. Furthermore, buffers must to added to the simpler decoder blocks 120 to match the overall delay to the slowest decoder block (i.e. the one with the most inputs).

Thus, to minimize aggregate decoder complexity, it is desirable to minimize the largest number of inputs to any given decoder block. In other words, it is desirable to have an even distribution of inputs to each decoder block 120. This is precisely what the Q-Gray code allows. Therefore, the Q-Gray code provides a means for minimal decoder circuit complexity which reduces dissipated power and increases achievable operational speeds.

Figures 1B and 1C illustrate exemplary embodiments for ADC's based on the 3-bit Q-Gray code (based on the BTOL in Eq. 14) and the 5-bit Q-Gray code (based on the BTOL in Eq. 15), respectively. Specifically, in Figure 1B, the A-bit decoder 110A takes as inputs logical "0" (from the seed value) and comparator outputs Vc1 and Vc5 since A occurs in the BTOL sequence in Eq. 14 at the first and fifth positions. Similarly, in Figure 1C, the A-bit decoder 110A takes as inputs logical "0" (from the seed value) and comparator outputs Vc1, Vc3, Vc5, Vc7, Vc9, and Vc21 since A occurs in the BTOL sequence in Eq. 15 at the first, third, fifth, seventh, ninth, and twenty-first positions.

Referring now to Figure 1D, this Figure illustrates a more specific realization of Figure 1A. In particular, it shows how the 4-input odd-parity functions are realized by using 3 XOR gates. The exemplary ADC circuit 100 illustrated in Figure 1D has been realized in a 2 μm GaAs HBT process and has been verified to be fully operational.

The circuit blocks used in Figure 1D can be implemented in a variety of circuit technologies and by a variety of design techniques by one skilled-in-the-art. One-skilled-in-the-art will realize that ADC's 100 with other bit resolutions can be similarly implemented in view of the teachings presented herein and the present art.

Another advantage of a Q-Gray code-based ADC is apparent from Figures 1A, 1B, and 1C. The decoder blocks 120 (i.e. odd-parity detector 110 followed by a latch 115 enclosed in a dashed box) are identical for each of the bits. The only difference is in the inputs assigned to each odd-parity detector. Thus, a single and simple decoder circuit can be designed, which can then be replicated for the other bit decoder blocks 120. This is in contrast to ADC's based on the conventional Gray code where each decoder block must be separately designed due to the high variation in the number of inputs to each block.

Referring now to Figure 2, this Figure illustrate a top-level schematic diagram of the exemplary ADC 100 shown in Figure 1D. Comparing this schematic to the block diagram of Figure 1D, it can be seen that the Vr1 to Vr15 comparator reference inputs of Figure 1D correspond to the Vre1 to Vre15 comparator inputs on the left-hand side of the schematic shown in Figure 2.

And the bit decoders and the associated comparators are absorbed within the circuit blocks labeled VC4b and VC3b. The voltage input is labeled Vin and is first buffered by the two buffering amplifiers, REFBUF (X58 and X59) circuit, before being routed to the comparator array, which is included within the VC3b and VC4b circuits (X54 through X57).

The outputs of the VC blocks are latched using 4 D-type flip-flop (DFF) circuit blocks. The latched outputs are then buffered to 50 Ω output impedance by a circuit block named OUTBUF2. One skilled in the art will realize that the functions of the DFF, BUF, INBUF, CLKBUF10U, DCDET, and OUTBUF2 circuit blocks can be realized in a variety of forms and representative circuit embodiments.

Referring now to Figure 3, this Figure illustrates the VC4b circuit that comprises four comparators, COMP (X1 through X4), and three exclusive-OR gates, XOR (X5 through X7). The VC circuits include the bit decoder and corresponding comparators, as required in Figure 1D, to determine a given bit's value.

Referring briefly to Figure 4, this Figure illustrates a simplified VC circuit. Referring to Figure 5, this Figure illustrates the REFBUF circuit that comprises an emitter follower amplifier. Figure 6 illustrates the IREF circuit used to bias the other circuits. And Figure 7 illustrates the circuit diagram of the comparator (COMP) circuit. Bit A has only three bit toggles and therefore a simplified VC circuit as shown in Figure 4 was used.

Figure 8 illustrates a simulation result for the exemplary ADC circuit of Figure 2. The input signal 802 is the trace labeled Vin at the top of the first graph 804. The top most level corresponds to level 15 of Table 3 and the lowest voltage level corresponds to level 0. A careful inspection of the outputs A, B, C, and D illustrated in Graphs 800A, 800B, 800C, and 800D demonstrate that the ADC's operation exactly matches the codes of Table 3 (notwithstanding the output delay of one symbol period, i.e. 333ps), as desired.

It should be understood that the foregoing relates only to illustrative embodiments of the present invention, and that numerous changes may be made therein without departing from the spirit and scope of the invention as defined by the following claims.

CLAIMS

What is claimed is:

1. An analog to digital converter comprising:

a plurality of comparators for receiving a multilevel signal generated according to a code for which the maximum number of bit-toggles incurred in a bit channel while sequentially traversing the code is minimized;

a plurality of decoder blocks coupled to comparators for decoding the multilevel signal, wherein each decoder block comprises an equal number of inputs, whereby the analog to digital converter has a simple design that reduces dissipated power and increases achievable operational speeds for communications.

2. The analog to digital converter of Claim 1, wherein the comparators can process a multilevel signal comprising non-uniformly spaced decoding thresholds.

3. The analog to digital converter of Claim 1, wherein the multilevel signal is decoded into bit channels for which the bit error rates are substantially evenly distributed.

4. The analog to digital converter of Claim 1, wherein each decoder block comprises a parity detector with an equal number of inputs.

5. The analog to digital converter of Claim 1, wherein each decoder block comprises a bank of identical parity detectors relative to another decoder block.

6. The analog to digital converter of Claim 1, wherein each comparator has an individually adjustable threshold level.

7. The analog to digital converter of Claim 1, wherein each comparator has an externally controllable threshold level.

8. An analog to digital converter comprising:

a plurality of comparators with adjustable thresholds for receiving a multilevel signal;

a plurality of decoder blocks coupled to the comparators for decoding the multilevel signal, wherein each decoder block comprises a plurality of identical parity detectors relative to another decoder block, whereby the analog to digital converter has a simple design that reduces dissipated power and increases communication speed.

9. The analog to digital converter of Claim 8, wherein the multilevel signal is decoded into bit channels for which the bit error rates are substantially evenly distributed.

10. The analog to digital converter of Claim 8, wherein the multilevel signal is decoded according to a code for which the maximum number of bit-toggles incurred in a bit channel while sequentially traversing the code is minimized.

11. The analog to digital converter of Claim 8, wherein each decoder block comprises an equal number of inputs.

12. The analog to digital converter of Claim 8, wherein each parity detector has an equal number of inputs.

13. The analog to digital converter of Claim 8, wherein each comparator has an individually adjustable threshold level.

14. The analog to digital converter of Claim 8, wherein each comparator has an externally controllable threshold level.

15. An analog to digital converter comprising:

a plurality of comparators for receiving a multilevel signal generated by a code for which the bit error rate is substantially evenly distributed across each bit channel when decoded;

a plurality of decoder blocks coupled to the comparators for decoding the multilevel signal, wherein each decoder block comprises an equal number of parity detectors, whereby the analog to digital converter reduces power consumption.

16. The analog to digital converter of Claim 15, wherein the comparators can process a multilevel signal comprising non-uniformly spaced decoding thresholds.

17. The analog to digital converter of Claim 15, wherein the multilevel signal is decoded according to a code for which the maximum number of bit-toggles incurred in a bit channel while sequentially traversing the code is minimized.

18. The analog to digital converter of Claim 15, wherein each decoder block comprises an equal number of inputs.

19. The analog to digital converter of Claim 15, wherein each parity detector has an equal number of inputs.

20. The analog to digital converter of Claim 15, wherein each comparator has an individually adjustable threshold level.

21. The analog to digital converter of Claim 15, wherein each comparator has an externally controllable threshold level.

22. An analog to digital converter comprising:

a plurality of comparators for receiving a multilevel signal, each comparator having an individually adjustable threshold level;

a plurality of decoder blocks coupled to the comparators for decoding the multilevel signal, wherein each decoder block comprises identical hardware relative to another decoder block, whereby the analog to digital converter has a simple design that increases communication speeds.

23. The analog to digital converter of Claim 22, wherein the multilevel signal is decoded according to a code for which the maximum number of bit-toggles incurred in a bit channel while sequentially traversing the code is minimized.

24. The analog to digital converter of Claim 22, wherein the comparators can process a multilevel signal comprising non-uniformly spaced decoding thresholds.

25. The analog to digital converter of Claim 22, wherein the hardware comprises parity detectors.

26. The analog to digital converter of Claim 25, wherein each parity detector has an equal number of inputs.

27. The analog to digital converter of Claim 22, wherein each decoder block comprises an equal number of inputs.

28. The analog to digital converter of Claim 22, wherein each comparator has an externally controllable threshold level.

29. The analog to digital converter of Claim 22, wherein inputs are evenly distributed to each decoder block.

30. An analog to digital converter comprising:

a plurality of comparators for receiving a multilevel signal generated according to a three bit Gray code derived from a bit translation operating list comprising ABCBABC with an operator comprising B and the operator returning an original seed from a last word in the three bit Gray code;

a plurality of decoder blocks coupled to comparators for decoding the multilevel signal, wherein each decoder block comprises an equal number of inputs.

31. The analog to digital converter of Claim 30, wherein the code producing the multilevel signal is further derived by cyclically rotating the bit translation operating list.

32. The analog to digital converter of Claim 30, wherein the code producing the multilevel signal is further derived by interchanging bit channel assignments.

33. An analog to digital converter comprising:

a plurality of comparators for receiving a multilevel signal generated according to a four bit Gray code derived from a bit translation operating list comprising CBCADABCBADBDCD with an operator comprising A and the operator returning an original seed from a last word in the four bit Gray code;

a plurality of decoder blocks coupled to comparators for decoding the multilevel signal, wherein each decoder block comprises an equal number of inputs.

34. The analog to digital converter of Claim 33, wherein the code producing the multilevel signal is further derived by cyclically rotating the bit translation operating list.

35. The analog to digital converter of Claim 33, wherein the code producing the multilevel signal is further derived by interchanging bit channel assignments.

36. An analog to digital converter comprising:

a plurality of comparators for receiving a multilevel signal generated according to a five bit Gray code derived from a bit translation operating list comprising ABACABADABEBDCDEBEDCACDECEBDCDE, with an operator comprising D and the operator returning an original seed from a last word in the five bit Gray code;

a plurality of decoder blocks coupled to comparators for decoding the multilevel signal, wherein each decoder block comprises an equal number of inputs.

37. The analog to digital converter of Claim 36, wherein the code producing the multilevel signal is further derived by cyclically rotating the bit translation operating list.

38. The analog to digital converter of Claim 36, wherein the code producing the multilevel signal is further derived by interchanging bit channel assignments.

39. A method for high speed communications comprising:

receiving data;

modulating the data to produce a multilevel signal according to a three bit Gray code derived from a bit translation operating list comprising ABCBABC with an operator comprising B and the operator returning an original seed from a last word in the three bit Gray code;

receiving the multilevel signal; and

converting the multilevel signal to a set of binary signals with decoder blocks, wherein each decoder block comprises an equal number of inputs.

40. The method of Claim 39, further comprising deriving the multilevel signal by cyclically rotating the bit translation operating list to yield an alternate code.

41. The method of Claim 39, further comprising deriving the multilevel signal by interchanging bit channel assignments.

42. A method for high speed communications comprising:

receiving data;

modulating the data to produce a multilevel signal according to a four bit Gray code derived from a bit translation operating list comprising CBCADABCBADBDCD with an operator comprising A and the operator returning an original seed from a last word in the four bit Gray code;

receiving the multilevel signal; and

converting the multilevel analog signal to a set of binary signals with decoder blocks, wherein each decoder block comprises an equal number of inputs.

43. The method of Claim 42, further comprising deriving the multilevel signal by cyclically rotating the bit translation operating list to yield an alternate code.

44. The method of Claim 42, further comprising deriving the multilevel signal by interchanging bit channel assignments.

45. A method for high speed communications comprising:

receiving data;

modulating the data to produce a multilevel signal to produce a five bit Gray code derived from a bit translation operating list comprising ABACABADABEBDCDEBEDCACDECEBDCDE, with an operator comprising D and the operator returning an original seed from a last word in the five bit Gray code;

receiving the multilevel signal; and

converting the multilevel signal to a set of binary signals with decoder blocks, wherein each decoder block comprises an equal number of inputs.

46. The method of Claim 45, further comprising deriving the multilevel signal by cyclically rotating the bit translation operating list to yield an alternate code.

47. The method of Claim 45, further comprising deriving the multilevel signal by interchanging bit channel assignments.

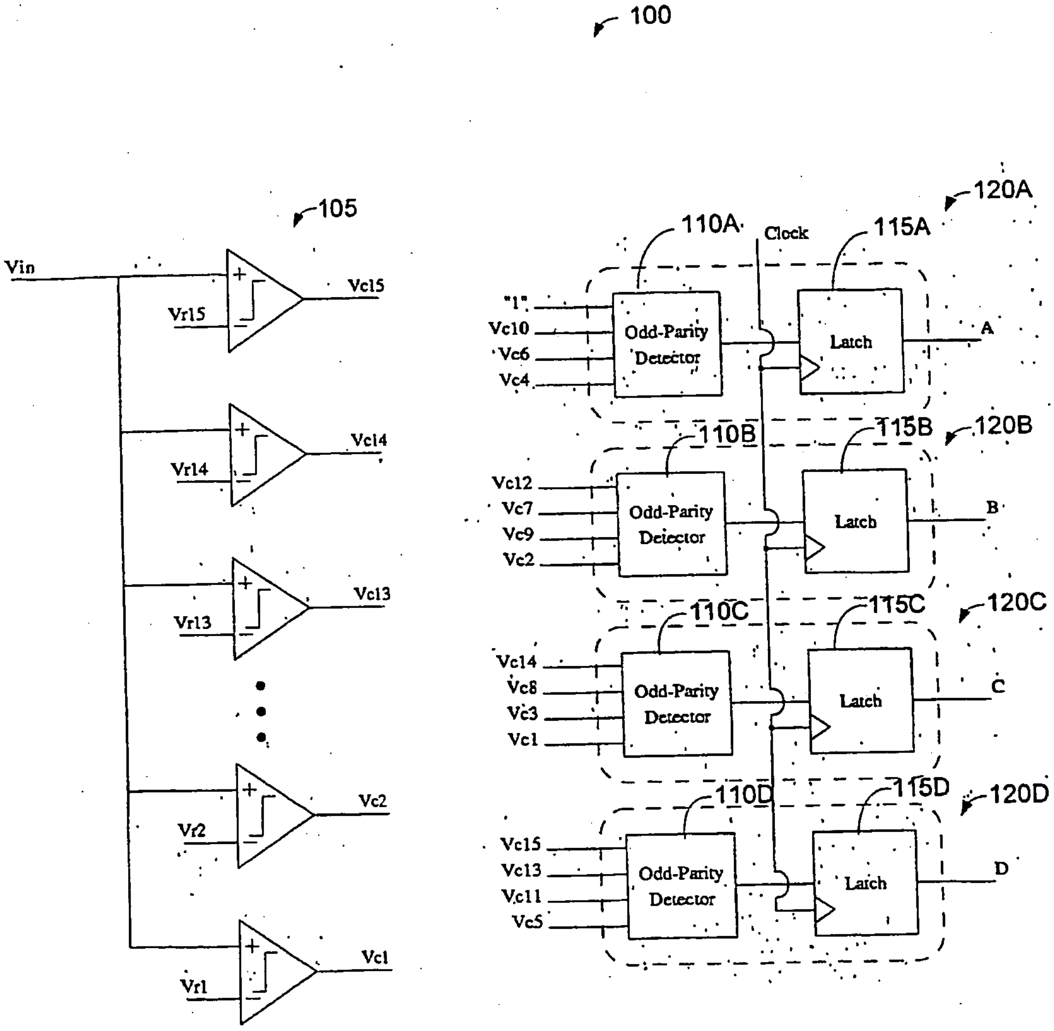

100
105
Vin
Vr15
Vc15
Vr14
Vc14
Vr13
Vc13
Vr2
Vc2
Vr1
Vc1
110A
Clock
115A
120A
"1"
Vc10
Vc6
Vc4
Odd-Parity Detector
Latch
A
110B
115B
120B
Vc12
Vc7
Vc9
Vc2
Odd-Parity Detector
Latch
B
110C
115C
120C
Vc14
Vc8
Vc3
Vc1
Odd-Parity Detector
Latch
C
110D
115D
120D
Vc15
Vc13
Vc11
Vc5
Odd-Parity Detector
Latch
D

FIG. 1A

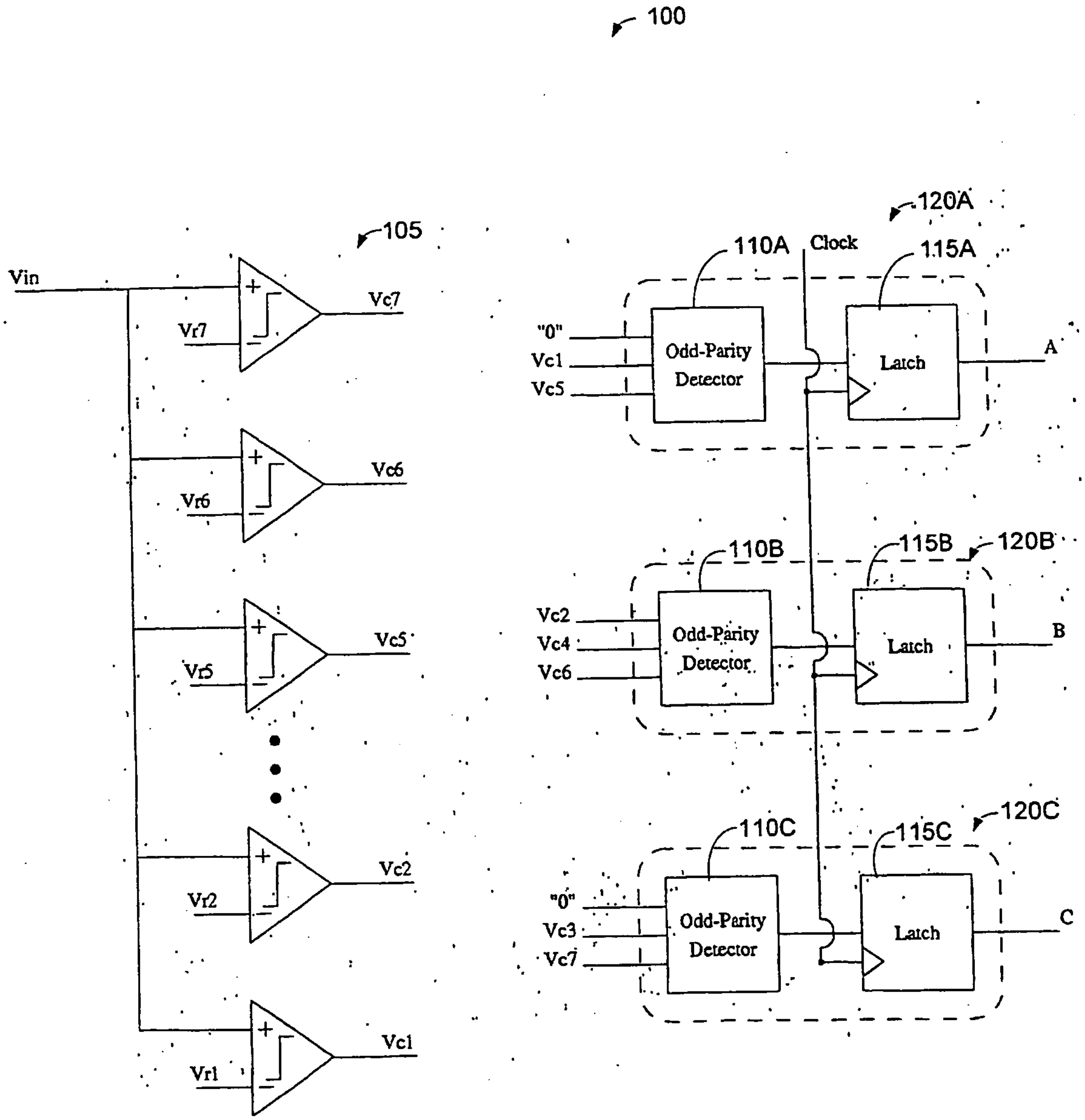

100
105
Vin
Vr7
Vc7
Vr6
Vc6
Vr5
Vc5
Vr2
Vc2
Vr1
Vc1
120A
110A
Clock
115A
"0"
Vc1
Vc5
Odd-Parity Detector
Latch
A
110B
115B
120B
Vc2
Vc4
Vc6
Odd-Parity Detector
Latch
B
110C
115C
120C
"0"
Vc3
Vc7
Odd-Parity Detector
Latch
C

FIG. 1B

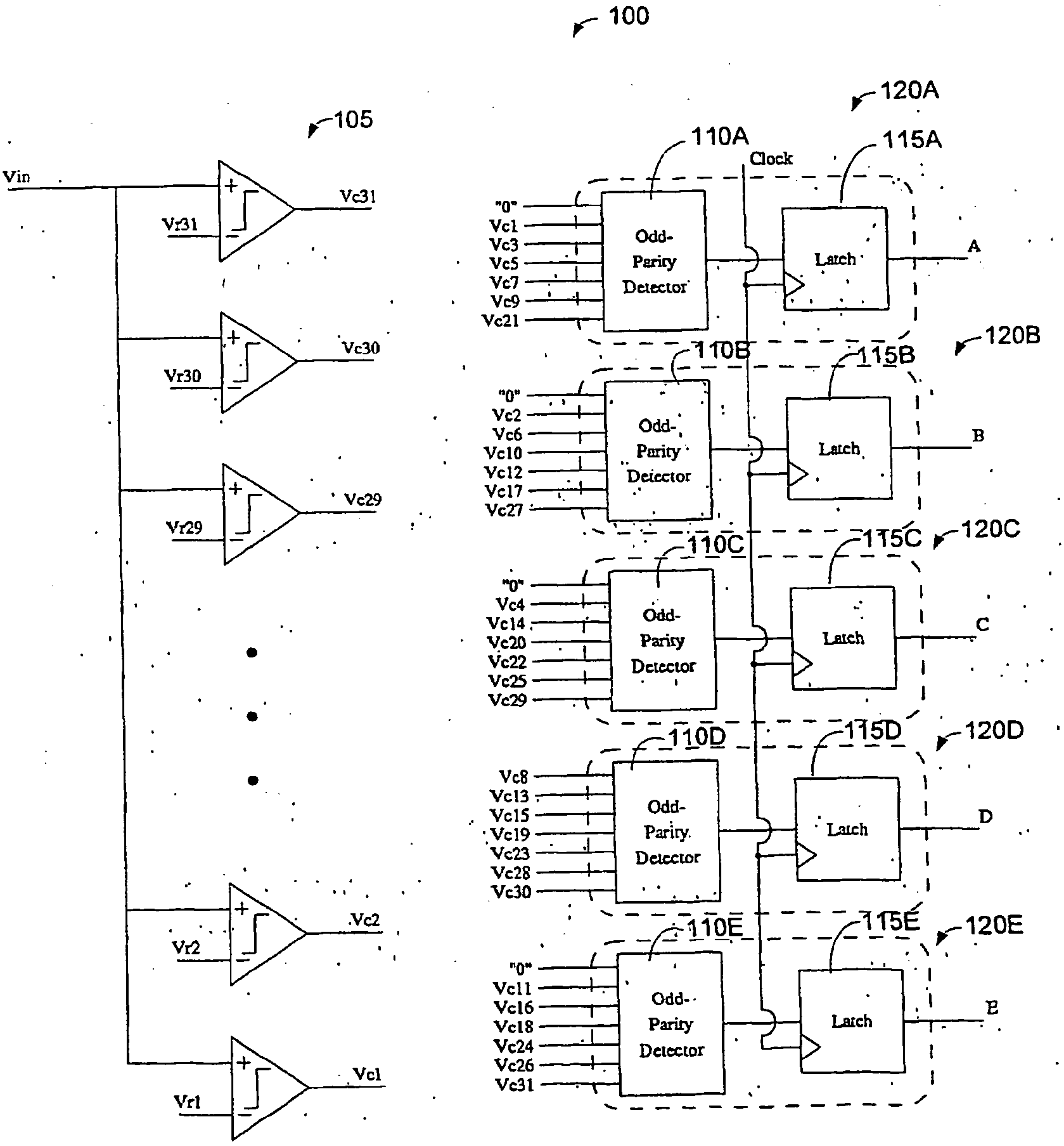

100
105
Vin
Vr31
Vc31
Vr30
Vc30
Vr29
Vc29
Vr2
Vc2
Vr1
Vc1
+
−
120A
110A
Clock
115A
"0"
Vc1
Vc3
Vc5
Vc7
Vc9
Vc21
Odd-Parity Detector
Latch
A
120B
110B
115B
"0"
Vc2
Vc6
Vc10
Vc12
Vc17
Vc27
Odd-Parity Detector
Latch
B
120C
110C
115C
"0"
Vc4
Vc14
Vc20
Vc22
Vc25
Vc29
Odd-Parity Detector
Latch
C
120D
110D
115D
Vc8
Vc13
Vc15
Vc19
Vc23
Vc28
Vc30
Odd-Parity Detector
Latch
D
120E
110E
115E
"0"
Vc11
Vc16
Vc18
Vc24
Vc26
Vc31
Odd-Parity Detector
Latch
E

FIG. 1C

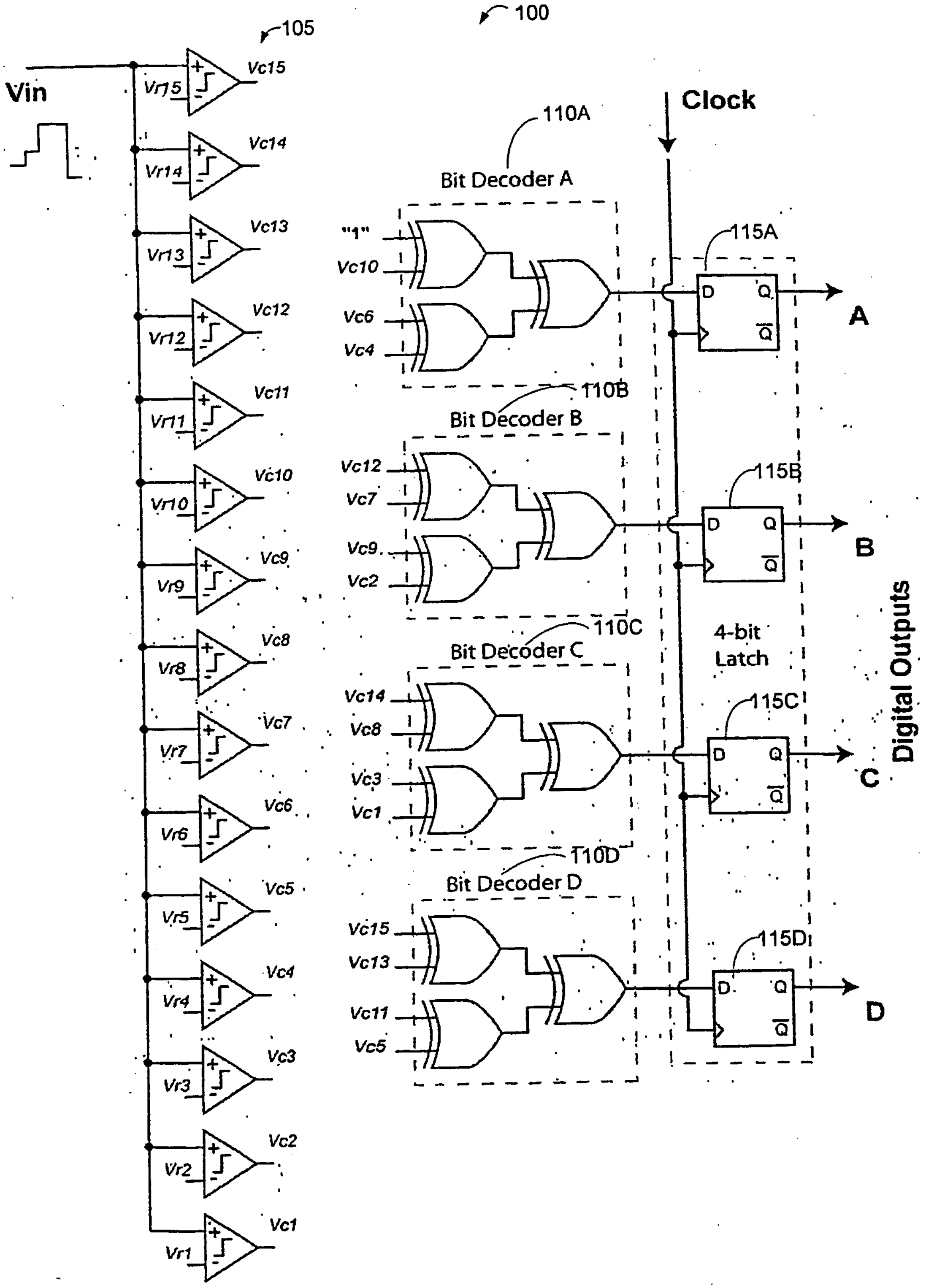

100
105
Vin
Vr15
Vc15
Vr14
Vc14
Vr13
Vc13
Vr12
Vc12
Vr11
Vc11
Vr10
Vc10
Vr9
Vc9
Vr8
Vc8
Vr7
Vc7
Vr6
Vc6
Vr5
Vc5
Vr4
Vc4
Vr3
Vc3
Vr2
Vc2
Vr1
Vc1
Clock
110A
Bit Decoder A
"1"
Vc10
Vc6
Vc4
110B
Bit Decoder B
Vc12
Vc7
Vc9
Vc2
110C
Bit Decoder C
Vc14
Vc8
Vc3
Vc1
110D
Bit Decoder D
Vc15
Vc13
Vc11
Vc5
115A
115B
115C
115D
D
Q
$\overline{Q}$
A
B
C
D
4-bit Latch
Digital Outputs

FIG. 1D

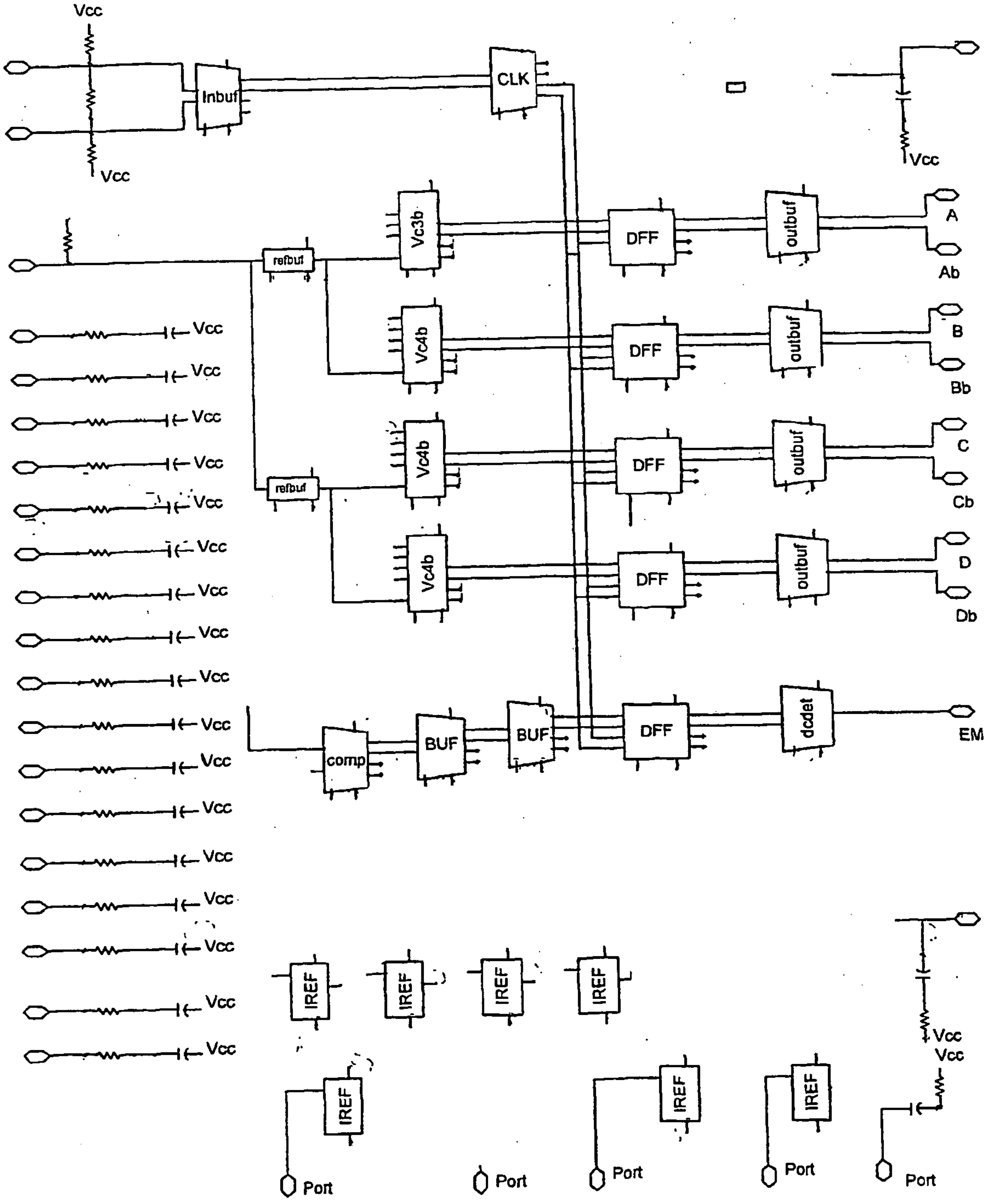

Vcc
Vcc
Inbuf
CLK
Vcc
refbuf
Vc3b
Vc4b
Vc4b
refbuf
Vc4b
DFF
DFF
DFF
DFF
outbuf
outbuf
outbuf
outbuf
A
Ab
B
Bb
C
Cb
D
Db
comp
BUF
BUF
DFF
dcdet
EM
IREF
IREF
IREF
IREF
IREF
IREF
IREF
Vcc
Vcc
Port
Port
Port
Port
Port

FIG. 2

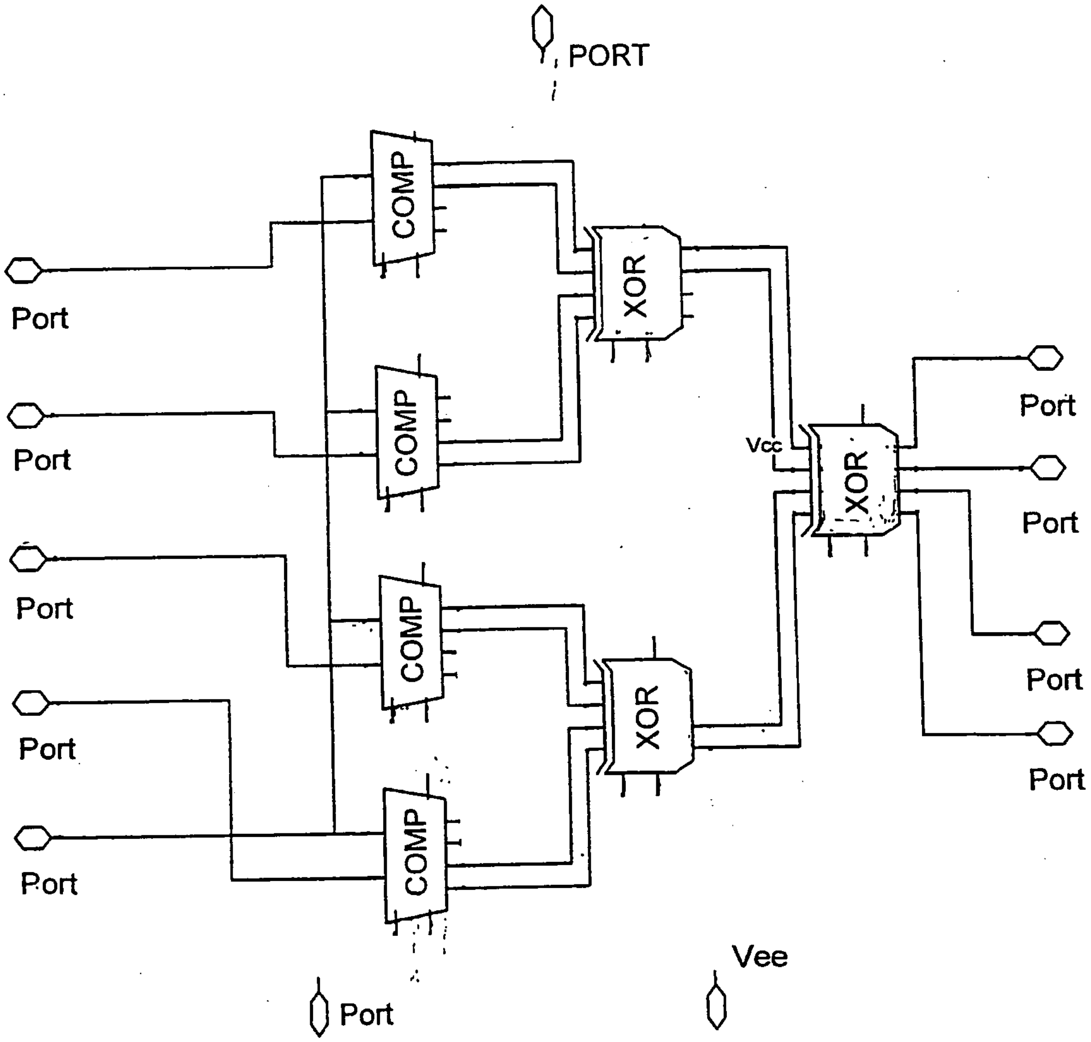

PORT
COMP
COMP
COMP
COMP
XOR
XOR
XOR
Port
Port
Port
Port
Port
Vcc
Port
Port
Port
Port
Vee
Port

FIG. 3

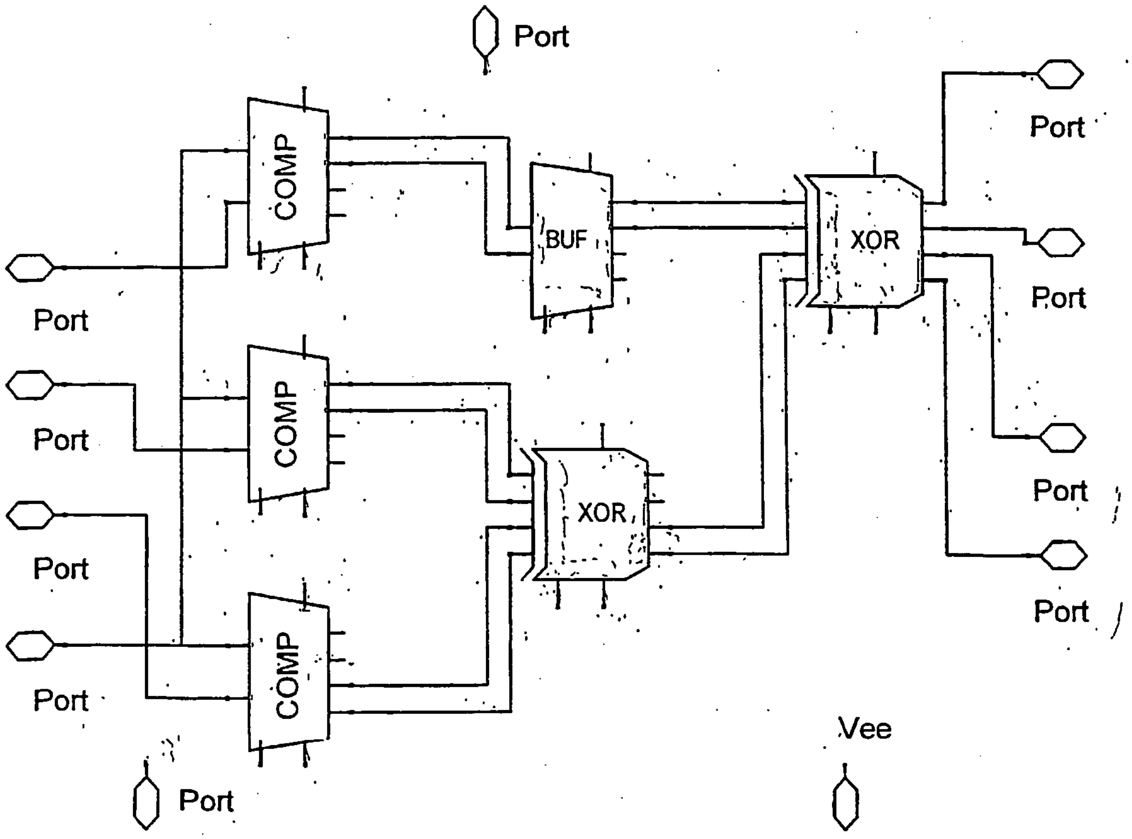

Port
COMP
BUF
XOR
Port
Port
COMP
XOR
Port
Port
Port
Port
Port
COMP
Port
Port
Port
Vee

FIG. 4

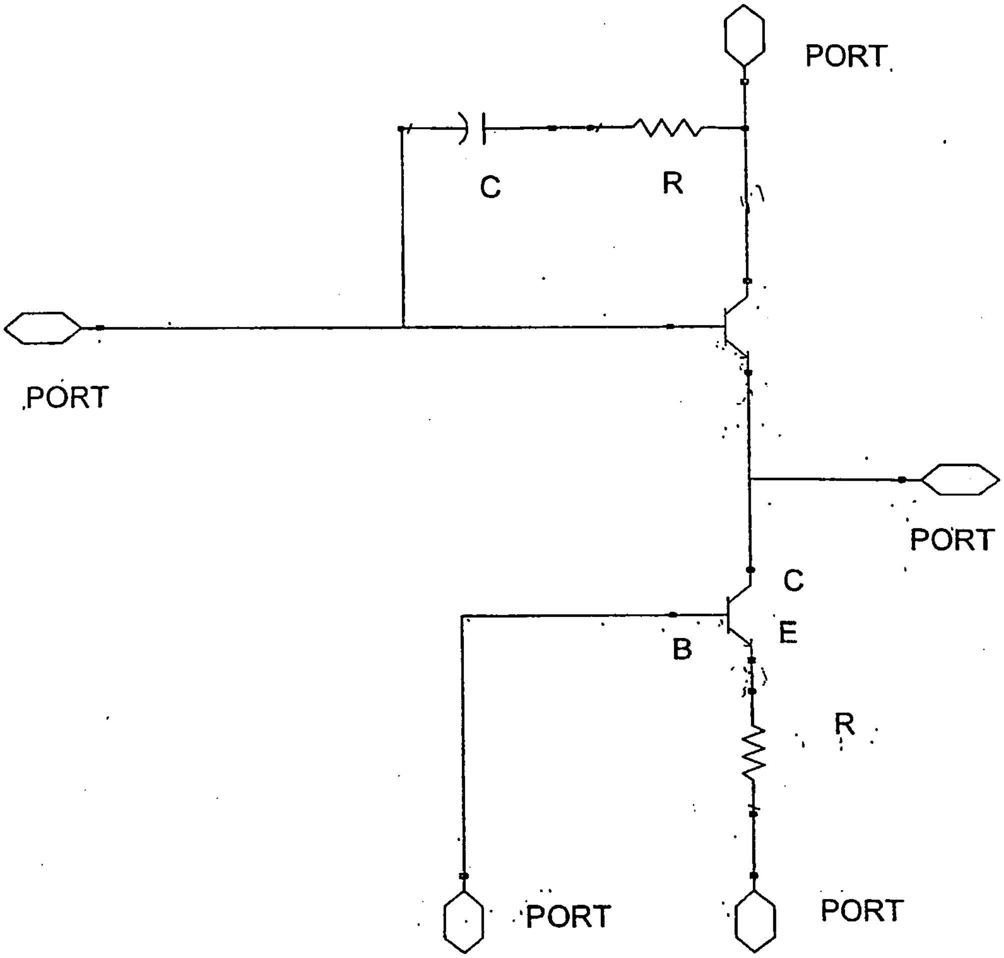


FIG. 5

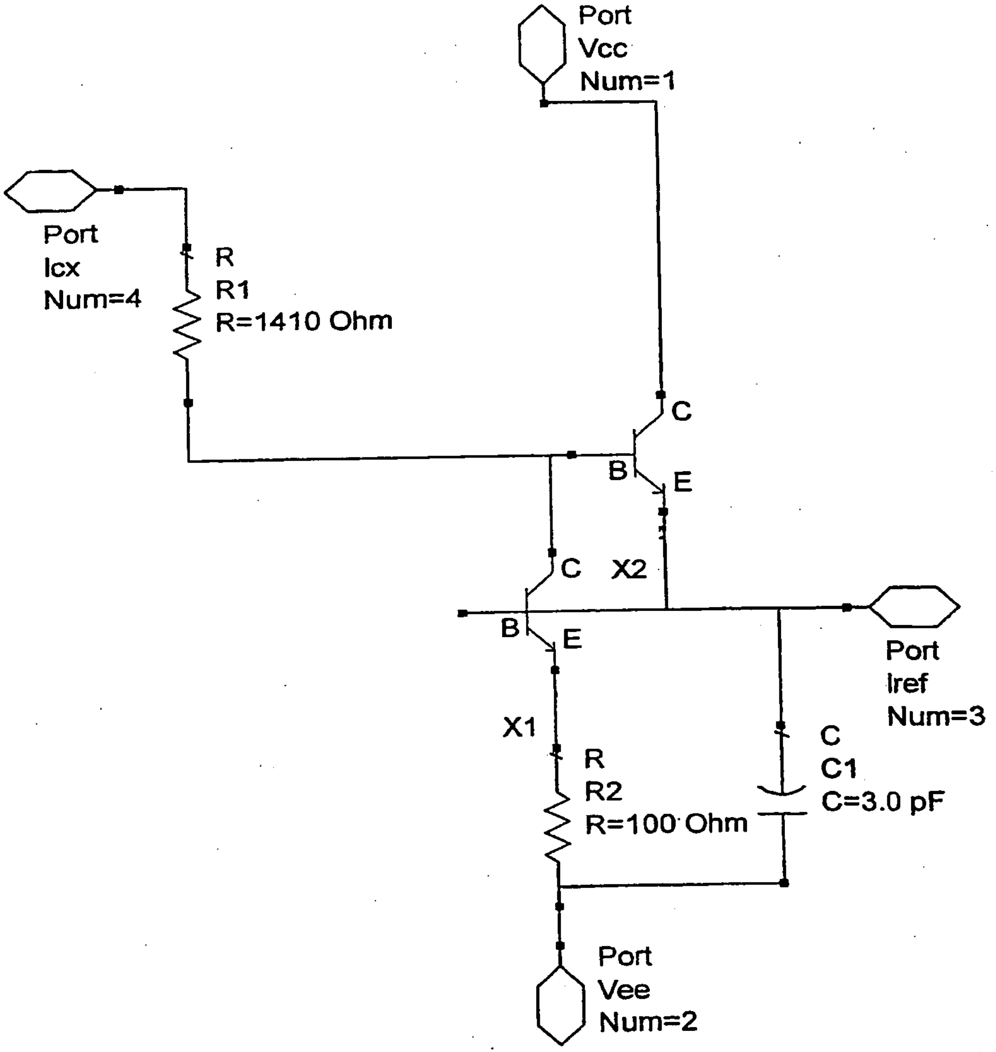

Port
Vcc
Num=1
Port
Icx
Num=4
R
R1
R=1410 Ohm
C
B
E
C
X2
B
E
Port
Iref
Num=3
X1
R
R2
R=100 Ohm
C
C1
C=3.0 pF
Port
Vee
Num=2

FIG. 6

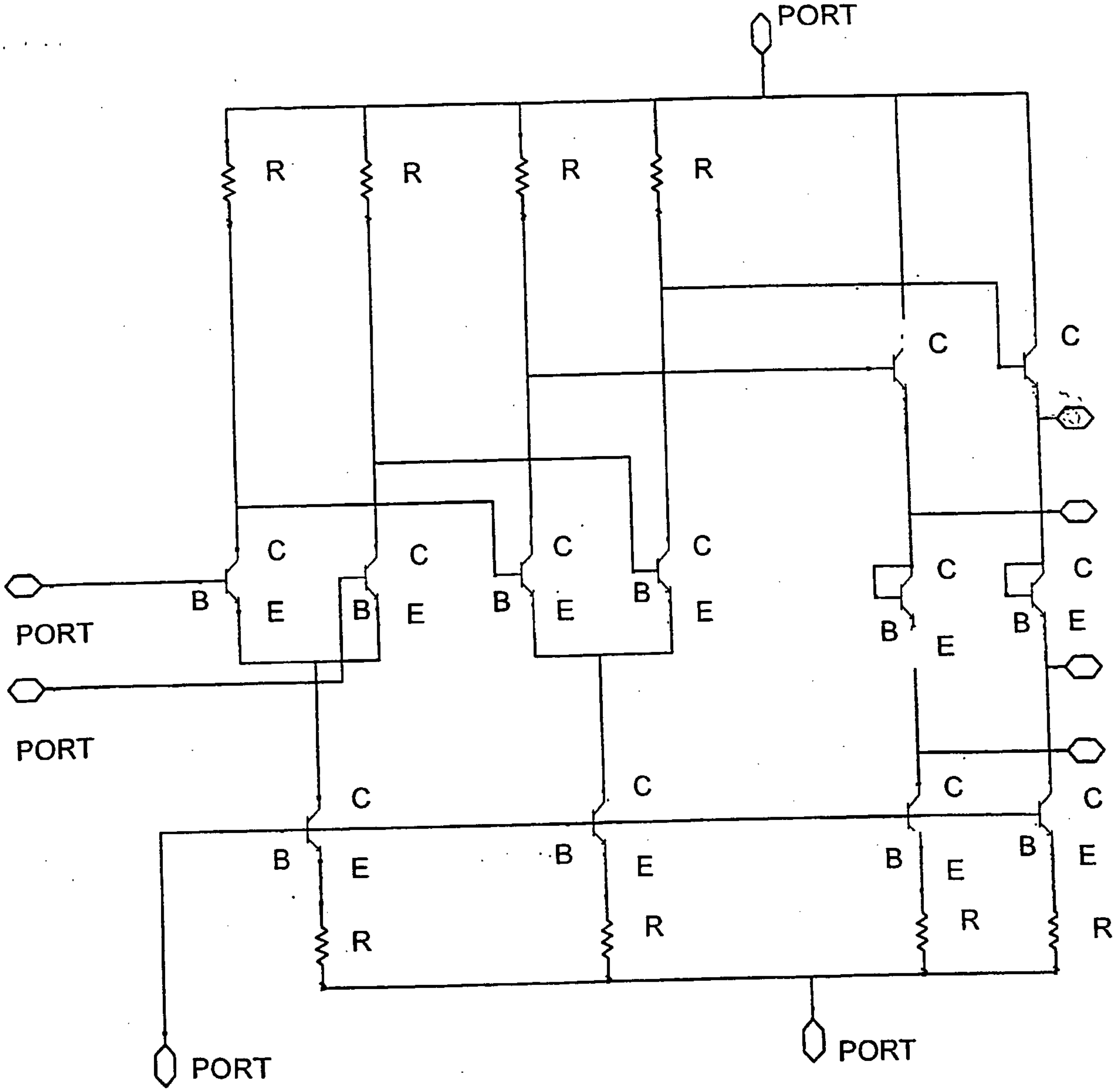

PORT
R
R
R
R
C
C
C
C
C
C
C
C
B
E
B
E
B
E
B
E
B
E
B
E
PORT
PORT
C
C
C
C
B
E
B
E
B
E
B
E
R
R
R
R
PORT
PORT

FIG. 7

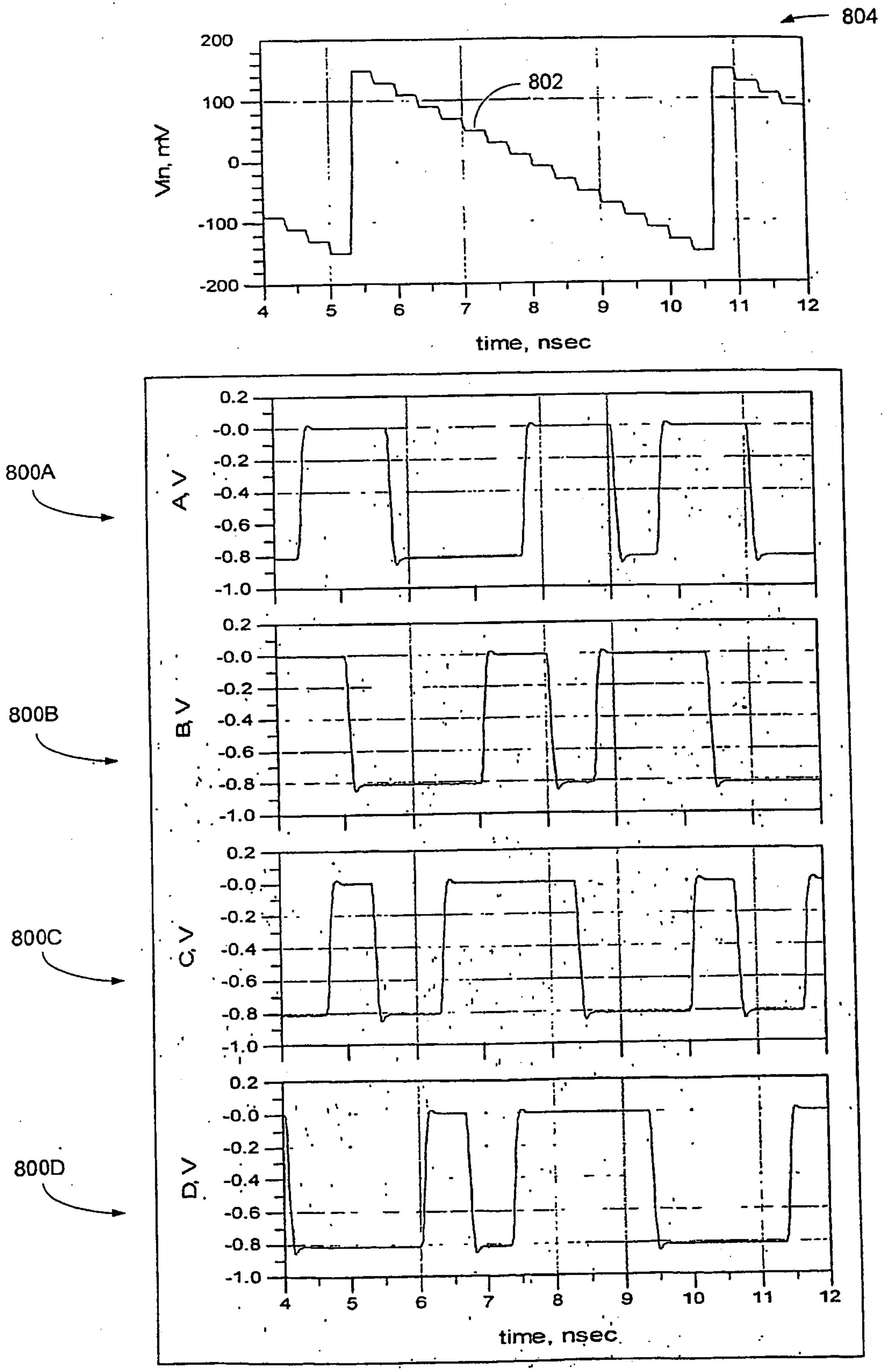

804
802
200
100
0
-100
-200
Vn, mV
4
5
6
7
8
9
10
11
12
time, nsec
800A
800B
800C
800D
A, V
B, V
C, V
D, V
0.2
-0.0
-0.2
-0.4
-0.6
-0.8
-1.0
time, nsec

FIG. 8